GRIDNET: A Highly Survivable Digital Communications Network Final Report, Phase I

Alan Mink
George G. Nacht
Alfred L. Koenig
Arthur W. Holt

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
Center for Computer Systems Engineering
Institute for Computer Sciences and Technology
Gaithersburg, MD 20899

December 1985

Issued May 1986

Sponsored by:
Defense Nuclear Agency
Washington, DC 20305
sk Code B99QMXPF
ork Unit 0002

NBSIR 86-3361

GRIDNET: A HIGHLY SURVIVABLE DIGITAL COMMUNICATIONS NETWORK FINAL REPORT, PHASE I

Alan Mink
George G. Nacht
Alfred L. Koenig
Arthur W. Holt

U.S. DEPARTMENT OF COMMERCE
National Bureau of Standards
Center for Computer Systems Engineering
Institute for Computer Sciences and Technology
Gaithersburg, MD 20899

December 1985

Issued May 1986

Sponsored by:
Defense Nuclear Agency
Washington, DC 20305
Task Code B99QMXPF
Work Unit 0002

U.S. DEPARTMENT OF COMMERCE, Malcolm Baldrige, *Secretary*
NATIONAL BUREAU OF STANDARDS, Ernest Ambler, *Director*

TABLE OF CONTENTS

ABSTRACT

GRIDNET is a highly reliable and survivable packet switched, wide area communication network that may consist of thousands of nodes and may span thousands of miles. The reliability of GRIDNET is based on redundant transmission of data via two distinct paths and bitwise comparison of the duplicate received data in addition to error detection codes. The survivability of GRIDNET is attributed to its intrinsic topology, which provides for a number of alternative paths between pairs of nodes. A feasibility prototype of a GRIDNET was proposed as a multi-phase research project. This report describes the design of the phase I GRIDNET prototype.

I. INTRODUCTION

GRIDNET [1,3,4,6] is a highly reliable and survivable packet switched, wide area communication network that may consist of thousands of nodes and may span thousands of miles. A GRIDNET network is a two tier structure. The lower level consists of a number of CROSSFIRE loops [2,5], see figure 1. The upper level is constructed by replicating and interconnecting CROSSFIRE loops in a regular hexagonal topology, see figure 2.

A CROSSFIRE loop is a polled, packet switched commmunication network using a dual loop topology. A CROSSFIRE loop is controlled by a single node called the Primary, all the other nodes on the loop are called Secondaries. The Primary continually polls every possible Secondary on the loop to determine its existence and its operability. If a Secondary node exists and is operational, it will respond to the Primary's poll with status information. Any packets to be exchanged between this Secondary and the Primary will then take place. Note that direct communication on a loop is only between a Secondary node and a Primary. All communication between two Secondary nodes is always indirect, with the Primary being the intermediary.

Each transmitted packet is simultaneously sent on both loops in opposite directions, one clockwise (CW) and the other counterclockwise (CCW). The packet flow is from the source node to the Primary, which removes all packets from the loop independent of where they originate. Thus a packet sent by the Primary is also removed by the Primary after completely traversing each loop, whereas a packet sent by a Secondary traverses only part of each loop and is removed when it reaches the Primary. Both copies of the packet (CW and CCW) are received by a node at different times due to the difference in path length of the two loops. When both copies have arrived they are first individually checked for errors and then bitwise compared against each other. If the CROSSFIRE loop is severed at any single location (even both CW and CCW loops) complete communication on the CROSSFIRE loop is still maintained, although only a single copy of the packet will be delivered. Severing of both loops at two separate locations is required to disrupt communications on a CROSSFIRE loop, and then only for the nodes that have become isolated from the Primary. This additional duplication of transmission on two loops and the comparison of data received on each loop contribute to the reliability of GRIDNET.

There are two classes of nodes on a CROSSFIRE loop, Gateways and non-Gateways. Gateway nodes have special privileges and responsibilities. They have the ability to become a Primary node, while a non-Gateway node always remains a Secondary node. There must be only one Primary at any time, but the Primaryship responsibility is periodically passed among Gateway nodes. When a Gateway node is not a Primary, it functions as a Secondary node. The Primary and Secondary activities of nodes are characteristics

of the Link Layer Communication protocol on a CROSSFIRE loop comprising the lower tier of the GRIDNET structure.

Gateways are also used to interconnect neighboring CROSSFIRE loops by conceptually splitting a Gateway in half and placing one half of the Gateway on one CROSSFIRE loop and the other half on the adjacent CROSSFIRE loop. Each half of the Gateway functions as an independent node on its respective CROSSFIRE loop, although both halves can readily communicate with each other in order to exchange information. The regular hexagonal interconnect topology of GRIDNET requires up to four Gateway nodes per interior CROSSFIRE loop, whereas boundary CROSSFIRE loops will have less than four since they connect to less than four other CROSSFIRE loops. Gateway nodes are responsible for routing within GRIDNET and maintaining information about the operability status of their neighboring CROSSFIRE loops. Thus, packets traverse GRIDNET by being passed between CROSSFIRE loops via Gateway nodes. To minimize operability information traffic, each Gateway only maintains operability information on its "local" neighborhood. Through simulation it has been determined that local operability information spanning a "2-neighborhood" (consisting of all CROSSFIRE loops within 2 hops of its own CROSSFIRE loop) yields good performance. These activities of Gateway nodes are characteristics of the Network Layer Communication protocol, thus forming the upper tier of the GRIDNET structure.

When packets are sent from one node to another, the first hop is to the current Primary on that CROSSFIRE loop, which is a Gateway. That Gateway checks the destination address of the packet. If the destination address is on that loop, the Gateway (being the Primary) will send the packet to the local destination node. If the destination address is not on that loop, the Gateway computes a route through GRIDNET (the interconnected CROSSFIRE loops) to the CROSSFIRE loop of the destination node. The Gateway then passes the packet to its other "half" which resides on the adjacent CROSSFIRE loop. The other half of the Gateway sends the packet (when it becomes Primary) to another Gateway on its CROSSFIRE loop which is specified in the route. That Gateway checks the specified route for any outages it is aware of and if necessary computes a new route. This Gateway then passes the packet to its other "half" which resides on yet another adjacent CROSSFIRE loop. This process continues until the packet arrives at a Gateway on the destination CROSSFIRE loop, which holds the packet until it becomes Primary and then delivers it to the destination node. A GATEWAY will hold a packet destined for another node on the same loop until it becomes PRIMARY, thus requiring only one transmission of that packet. The alternative is to send the packet to the current PRIMARY for forwarding to the destination node, which requires two transmissions of the packet (or more if PRIMARYSHIP changes prior to forwarding).

Since a Gateway only maintains operability information on its local neigborhood, it may compute a routing which contains outages. As the packet traverses the network it will arrive at a Gateway in the neighborhood of that outage. That Gateway, upon checking the route of the packet, will compute a new route to avoid the known outages in its neighborhood, again based on local vs. global information. This may occur a number of times as the packet makes its way towards its destination. Any Gateway can compute a new route to any destination without the use of routing tables because of the regular hexagonal topology and a consistent node numbering scheme of GRIDNET.

Multiple paths between any pair of nodes is another feature of the GRIDNET topology, and the number of paths increases as the network grows. As a result, the lack of global operability data does not affect on the ability to deliver a packet, it only affects on the efficiency of delivery and then only when there are outages. This degradation may be somewhat compensated for by the fact that the network will not be flooded by global distribution of operability information. Thus GRIDNET could sustain the loss of many CROSSFIRE loops but still always be able to find a path through to surviving sites, if one exists, without the use of routing tables and global information.

The GRIDNET project was divided into three phases. Phase I was to design and implement a single CROSSFIRE loop consisting of four nodes, a single Gateway (therefore Primary) and three non-Gateways (therefore Secondaries) communicating using the Link Layer GRIDNET protocol. Phase II was to implement two additional CROSSFIRE loops (for a total of 3) populated with 26 nodes, 8 Gateways and 18 Secondaries, thus forming a 3-loop GRIDNET, and the associated Network Layer Protocol. Phase III was to implement a larger GRIDNET consisting of 100 nodes distibuted amoung 20 loops and to conduct performance experiments on that configuration. This size was determined to be the smallest GRIDNET possible to fully test the performance, routing, and survivability of the GRIDNET concept.

Phase I consisted of four objectives: (1) design and implement a CROSSFIRE loop and its associated Physical and Link layer communications protocols, (2) develop communications between the CROSSFIRE nodes and a development system to download code and access loop information, (3) demonstrate the ability of the CROSSFIRE loop to detect outages, and (4) demonstrate the ability of the CROSSFIRE loop to locate and diagnose the outage. Some basic parameters had to be established to initiate the design of a CROSSFIRE loop, such as transmission media, transmission speed, and communication protocols. The transmission media of the CROSSFIRE loop was to be fiber optics including sophisticated switches that would by-pass any node that was down. Transmission on the media was to be serial data at 1 mega-bit per second. The Link layer protocol would follow the American National Standard Institute (ANSI) Advanced Data Communication Control Procedures (ADCCP) [7].

This report describes the design of phase I of GRIDNET. GRIDNET development was divided into three major efforts: (1) the architecture that would comprise a network node, (2) the software/micro-code to implement the communication protocols within each node, and (3) the fiber optics for the transmission media and by-pass switches. This report is organized along that same line.

II. Node Architecture

The function of a node is to provide an interface between a user or host and the network via a set of protocols which allow the transfer of messages and status information. The protocols for this design required an Application specific layer and the lower layers (Physical, Link, and Network), but no middle layers (Presentation, Session, and Transport) of the Open System Interconnection Reference Model [8]. The Application layer would implement the application specific dialog to a user or a host. The middle protocol layers would be left null for possible future implementation.

The lower protocol layers are the target of this design effort. The architecture of a node was divided into two modules along the lines of the protocols. One module, the Front-End (FE), would implement the Physical and Link layer protocols, which involved time critical operations. The other module, the Single Board Computer (SBC), would implement the remaining upper protocol layers. This division of labor is expected to increase performance by reducing the turnaround time for the exchange of packets on the network and minimizing backplane utilization.

The general configuration of a node is shown in figure 3. Commercially available components were used whenever possible in its construction. The CROSSFIRE transmission media (fiber optics) are connected to the FE via a specially designed electro-optics interface. The FE itself is a specially designed piece of hardware which is controlled by an 8x305 8-bit, bipolar, Shottky microprocessor and its associated microcode. The SBC is a 68000-based single board computer with 128K bytes of dual-ported memory. The SBC and the FE are tied together via an IEEE-796 backplane. The SBC views the FE as a dedicated special device with direct memory access (DMA) to its memory. Thus messages and data are passed between the SBC and the FE through the SBC's dual-port memory, which the FE accesses across the IEEE-796 bus via its DMA. The user or host is connected to the SBC via either a RS-232 serial interface or a parallel interface. There is also a time distribution system attached to the IEEE-796 backplane. This was to provide a uniform timestamp at each node for the collection of performance statistics on this feasibility

prototype. This time system is not meant for an operational environment.

Front End Architecture

The organization of the FE is shown in figure 4. The communication between the two processors is across the IEEE-796 bus through a "slave" and a "master" interface. The "slave" interface is only used to download micro-code for the FE. This micro-code is stored in a 2K x 48 bit RAM which, once loaded, is then used in a read-only mode (i.e., the 8x305 cannot modify its instruction memory). A micro-instruction is 40 bits wide, 16 bits are used for the 8x305 instruction and 24 bits are used to control the devices and data flow within the FE. The "master" IEEE-796 bus interface allows the FE to assume control of the IEEE-796 bus for the purpose of DMA transfers of information to and from the SBC's memory. Within the FE there is a local 8-bit bus controlled by the 8x305 and the micro-code. In addition to the FE instruction memory there is also a local 4K byte data RAM. This provides storage for 8x305 variables and three packets, two outgoing packets and one incoming (overflow) packet. Special address generation logic was necessary since the 8x305 is an 8-bit processor and 12 bits are required to access the local 4K data RAM. This address logic was organized around a set of base registers which are dynamically writeable by the 8x305. The micro-instruction would reference one of these base registers and in addition specify 1 of 2 values to be added to the base register. The 2 possible values were (1) a constant specified in the micro-instruction or (2) a dedicated counter with optional reset and post-increment capabilities.

The I/O portion of the FE is detailed in figure 5. The Link protocol layer followed the ADCCP bit-oriented protocol [7], which allowed the use of a commercially available ADCCP chip. This chip would handle the capture and transmission of a packet as follows. When transmitting, the chip accepts the packet a byte at a time and sends it out serially at the synchronous transmission speed, while adding the leading and trailing flags that bound a packet as well as computing and appending the Cyclic Redundancy Check (CRC) error check code. When receiving, the chip accepts the packet serially at the synchronous transmission speed and relays it a byte at a time, while stripping off the leading and trailing flags as well as computing the CRC and checking it against the one received. Also, address recognition capabilities allow the chip to selectively receive only packets with a specified address.

Because of the CROSSFIRE configuration, two chips were needed, one for each loop. For transmission, the 8x305 under program control would place the chips into transmit mode and then send the packet a byte at a time to both chips simultaneously. This was a time critical operation, since once started the packet transmission was synchronous. The chips would serialize the

packet and pass it through an electro-optical (E/O) interface into the fiber optic transmission media.

The receiving side of the I/O is somewhat more complex because of the additional GRIDNET requirement to bitwise compare both copies (CW and CCW) of the received packet. The difficulty arises from the difference in arrival times of the two packets due to the propagation delays through the two different transmission paths. The 8x305, under program control, would place the chips into receive mode. Incoming data, is received serially by the ADCCP chip through an electro-optical (E/O) interface from the fiber optic media. The two copies of the packet arrive and are processed by the chips at different times. Each packet is independently transferred from the chip, a byte at a time, to a first-in-first-out (FIFO) buffer in order to resynchronize them. When the corresponding byte of each packet is in the FIFO, they are removed, and compared against each other, setting an accumulative error latch for any mismatch, and then both bytes are transferred to receive buffers. The receive buffers, called ping-pong buffers, are actually four buffers organized into 2 pairs that function as a double buffer. One pair of buffers store both copies (CW and CCW) of the received packet, leaving the other pair available to store the next packet in case it arrives before the previous packet has been processed. Because of the GRIDNET polling protocol, double buffering is sufficient to insure that packets will not be lost. Both copies of the received packet must be stored since it is not known which one, if either, is correct until both packets have been received and the CRC and the content checks are done.

Two additional receive error conditions must be checked. When a packet starts to arrive on either loop a timer is started. This timer is set to the maximum propogation delay on a loop. If the timer expires before data is received on the other loop this implies there is a break or malfunction on the other loop. This constitutes a loop error. There are also timers to detect byte dropouts during synchronous reception. All receiver error indicators are accessible via the local bus by the 8x305.

Each of the FE's for phase I has been built on two wirewrap boards. This was necessitated because of the complexity and quantity of integrated circuits required. One board contains the IEEE-796 bus communication circuitry, the 8x305 and its micro-code instruction RAM with associated decoding logic, and the local data RAM and associated address logic. The other board contains the I/O functions, timers, the ADCCP chips, and other receiver/transmitter circuitry. These two boards are tightly coupled in that the communication between them is via a dedicated cable.

Both the 68000 based SBC and the 8x305 controlled FE can become master of the IEEE-796 bus. A daisy chain serial priority scheme was used to resolve bus contention. The IEEE-796 backplane being used orders priority by the physical location of the master

within the backplane. The highest priority master is located at the bottom slot in the cardcage. Since the FE handles more time critical information, the FE processor board (the one with the IEEE-796 bus communication) is placed at the bottom (highest priority) of the cardcage, while the SBC is next above it (second in priority). The FE I/O board is last in priority being placed above the SBC.

Gateway Architecture

A Gateway is a privileged node that connects two adjacent CROSSFIRE loops and conceptually consists of two halves, one on each loop. Thus a Gateway was constructed as two nodes (a node consists of an SBC and an FE) on a common (IEEE-796) backplane with a large shared memory, figure 6. This arrangement was selected as part of the modular design of GRIDNET to minimize the architectual components requiring maintenance. Some extra features had to be added to the hardware and micro-code in order to handle the additional functions of a Gateway, but the benefit of a smaller parts count and common code at each node outweighted the incremental cost.

Each half of the Gateway can communicate with the other half by exchanging messages through the shared memory. In this manner packets can be passed from one half of the Gateway to the other half, thus transfering a packet from one loop to another. Similarly status information on the operability of each loop can be exchanged. By constructing a Gateway in this manner an additional efficiency in packet handling can be realized from its reception on one loop to its retransmission on the other. Instead of three packet transfers, one from the receiving FE to its SBC for processing, a second from one SBC to the other SBC constituting a loop transfer, and a third from that SBC to its FE for retransmission, only two packet transfers are required in the current Gateway organization. A packet which is received by one FE is transfered directly to the shared memory where both SBCs can process it and then only a second transfer is needed to the other FE for retransmission.

III. Node Software

A GRIDNET node consists of two processing elements: (1) the Front-End (FE) and (2) the SBC. The SBC is the central control processor. It is responsible for initializing and coordinating the FE, handling the communication protocols from the network layer and above, collecting operational statistics, generating traffic, and providing an external maintenance, display, and control interface. The FE appears to the SBC as a special

purpose device which has DMA access to the SBC's memory, but is under the SBC's control. The primary function of the FE is to offload the SBC by handling the Link and Physical layer protocols.

The FE monitors the network and captures any packets addressed to it, processes those packets and formulates the required responses. The only information that is passed to the SBC, other than epoch occurrences for statistics and state changes, are data packets and XIDs (information required by the higher level protocols). The FE also monitors requests from the SBC, processes them and informs the SBC of the result. The predominant SBC request is to transmit a packet. The FE takes the packet (via its DMA access) and proceeds with the activities required to transmit the packet and on completion inform the SBC of the results. Thus the FE handles the bulk of the interactions with the network without burdening the SBC.

Node Operations

The normal operation of a node is as follows. On power up the SBC, the FE, and all other devices on the node are forced into an initial state. The SBC initial state forces it to begin executing code which resides on a PROM. This PROM contains a minimal monitor provided by the manufacturer, through which a link can be established with our development processor to download the SBC and FE code. The SBC, by executing its code, starts the operating system kernel which initializes the various SBC system and application processes. One of these processes loads the FE code into the FE (via the IEEE-796 bus) and then instructs the FE to begin executing that code. At this point the SBC and the FE are executing independently of each other. The FE then continuously monitors the network and waits for an event to occur (e.g., a packet arrival or an SBC request). When an event does occur, the FE processes that event and, if necessary, provides the SBC with the information resulting from that event.

Communication between the FE and the SBC is provided through common memory (the SBC memory is dual ported and the second port is externally accessible via the IEEE-796 bus), augmented by an interrupt from the FE to the SBC. Two communications areas within the SBC's memory are reserved by the SBC kernel, one for general operations (the Comm area) and the other for packet I/O operations (the I/O Register area). The SBC places a list of addresses into the Comm area and then "hard codes" the address of the Comm area into the FE code just prior to downloading it into the FE. On initialization the FE accesses the Comm area using the "hard-coded" address provided by the SBC and acquires the list of other addresses. Mutual exclusion between the FE and SBC for these memory areas are handled by a set of software locks.

When the SBC has a request for the FE, it accesses the proper communication area and sets the request. The FE is

continuously polling these communication areas and processes all new requests when it sees them. The FE provides two levels of acknowledgment to these requests. First, the FE upon accepting the request clears that request in the communication area. This allows the SBC to place additonal requests without waiting for prior requests to complete. Second, the FE upon completion of the request accesses the proper communication areas and leaves a status message for the SBC. The FE then interrupts the SBC to alert it to the existence of the status information. The SBC is interrupt driven and does not poll devices.

When an event occurs for which the FE must notify the SBC, the FE processes that event, and then places a status message in the proper communications area of the SBC's memory and then interrupts the SBC to alert it to the existence of this information. For example, when a data packet arrives, the FE copies the data packet to a receive packet buffer in the SBC's memory., The FE then leaves a status message in the I/O Register area of the SBC's memory stating a packet of length x has been placed in the current receive packet buffer, and then interrupts the SBC to alert it to the existence of this information.

Single Board Computer Code

The SBC code is organized as modules that can be divided into three classes: (1) the kernel, (2) system processes, and (3) application processes. The kernel is a streamlined operating system. It handles process scheduling, fields device interrupts, and provides a set of system functions available through supervisory (trap) calls. System processes are high priority processes which need access to the priviledged instruction set and are handled in a last-in-first-out priority order. Application processes are low priority processes that do not require access to the priviledged instruction set and do not require any special handling order.

On power up, the kernel clears the memory area allocated for data, and establishes the vector addresses for interrupts and a jump table for traps. Then a set of initialization functions are invoked, one for each device (i.e., the clock, the terminal, and the FE). These functions set the devices to the desired initial state and initialize any associated data structures. The FE initialization function modifies the FE code with the address of the communication area, and then it downloads the FE code into the FE. Once the devices are initialized, the kernel builds the process table, marking each process ready to run, and allocating stack space for each process as well as for itself. Initialization is completed by invoking the scheduler function, which begins the execution of the various processes. Some processes are only applicable to Gateway nodes, although their code is included in each node for uniformity. For a non-Gateway node these processes are initiated but sleep forever, unactivated by nonexistent events at those nodes.

The system processes consist of: (1) the terminal input process, (2) the terminal output process, (3) the packet receive process, (4) the packet send process, (5) the statistics gathering process, and (6) the node status process. The terminal input and output processes handle the read and write functions to the local terminal. This includes queueing of the I/O streams to maintain arrival order and buffering to offset the time differential between I/O producers and consumers. The packet send process accepts a complete packet ready to transmit, enqueues it on the proper destination queue, and depending on the status of the node (non-Gateway, Gateway, or Primary), follows the GRIDNET protocol to transmit those packets to their specified destinations. The packet receive process waits for a packet to arrive on the network, determines its destination and then processes it accordingly. If the received packet's destination is local, that packet is consummed locally. Currently this optionally prints the packet on the terminal and then discards it. In the future the packet will be sent to a user/host process for application specific processing. If the packet's destination is not local, then that packet is sent to the router process to determine its next destination. The statistics gathering process accumulates the information recorded on certain epochs for later reduction and statistical analysis. This information consists of mostly time stamped events (e.g., packet arrivals, packet transmissions, etc.) and counters (e.g., number of packets sent and received, number of each type of error, etc.). The node status process is not active for non-Gateway nodes. It maintains information regarding the operability of its CROSSFIRE loop and the CROSSFIRE loops in its local (two) neighborhood.

The application processes consist of: (1) the user/maintenance process, (2) the automatic packet generation process, (3) the traffic generation process, and (4) the router process. The user/maintenance process provides the current external interface to a GRIDNET node. This process allows the user to peruse and modify any memory locations, view a variety of status information, and provide the ability to cause certain events to occur (e.g., send a packet to a destination, wakeup a process, etc.). The automatic packet generation process creates a packet of random length at any specified regular intervals and enqueues it for transmission by the traffic generation process. The traffic generation process dequeues any packet ready for transmission independent of who or how it was created and passes it to the send process when the send process can accept it for transmission. The router process receives all incoming packets not directed to its node, independent of its arrival mode (either from its own loop or the other half of the Gateway). The router process is not active for non-Gateway nodes. The router checks the packet's destination and routing from the network layer encapsulated information. The viability of that routing is checked based on 2-neighborhood local information, and if that path is blocked a new path is computed. The packet is then enqueued for transmission along the next hop in the specified

path (either to another node on the loop or to the other Gateway half).

Front End Code

The FE is a special purpose interface dedicated to handling the Link and Physical layer protocol for GRIDNET. The FE is managed by an 8x305 8 bit controller. This controller is fast (333 nsec/instruction) and supports user defined micro-code, but has a small instruction set (8 instructions). The user defined micro-code is an important feature since it is used to control the activity of the entire FE (i.e., the other devices that comprise the FE).

The FE code is divided into three major activities: (1) initialization, (2) sampling, and (3) actions. The FE is initialized once at start up. Then control is passed to the sampling operation which executes the polling function, waiting for an event to occur. When an event is detected the appropriate action is invoked to process that event. When the action is completed, control is returned to the sampling operation.

The initialization activity begins at power up. The FE is forced into, and remains in, a wait state. Some time later the SBC downloads the instruction memory of the 8x305 controller (via the IEEE-796 bus) which, upon completion, causes the 8x305 to begin executing those instructions starting at location zero. The 8x305 first initializes the devices on the FE, such as its own registers, the 8x320 (a set of base registers to point into its local data memory), its local data memory, the (ADCCP) protocol chips, and the IEEE-796 bus interface chips (to the SBC's memory). The 8x305 then accesses the Comm area in the SBC's memory using the address that was "hard-coded" into its code by the SBC before it downloaded the 8x305 instruction memory. The 8x305 continually samples that location waiting for the SBC to indicate that the SBC is operational, and then the 8x305 waits for the SBC to indicate that the 8x305 should become operational. The 8x305 then accesses the Comm area to acquire other necessary addresses in the SBC's memory (e.g., the I/O Register area). The 8x305 completes the initialization activities by readying the (ADCCP) protocol chips with the node address and the desired network operational characteristics and then invokes the sampling operation.

The 8x305 (and thus the FE) is a polling device since there are no facilities to field and respond to interrupts. The sampling operation, waiting for an event to occur, carries out that polling function. The sampling operation polls the I/O Register area, the Comm area, the network, and a number of timers. The SBC places I/O requests in the I/O Register area. The most common event is the request to transmit a data packet. In the Comm area the SBC places operational directives to the FE, currently the SBC can request the FE to either become inactive

(for simulation purposes and for loading new software) or to become active. The network is polled to detect the event of an incoming packet. A number of timers are incorporated in the FE to detect exception events, such as receiver errors and response intervals. For example, loss of synchronization on received packets (byte error), no response on one loop (loop error), and no response to a packet transmission (node down, out of communications, or nonexistent). When an event occurs the appropriate action is invoked to process that event and, when completed, control is returned to the sampling operation.

All FE actions are the result of the occurence of some event detected during polling. In sampling the Comm area the events that may occur are requests from the SBC. Currently they consist of the SBC requesting the FE to either become inactive or to become active. The inactive state consists of reseting some internal state variables and then constantly polling the Comm area, ignoring all other events, until the active request is made by the SBC.

The events that may occur while sampling the I/O Register area are requests from the SBC. They consist of the SBC requesting the FE to either transmit a data packet or to load a received data packet into a packet buffer. To transmit a data packet, the SBC places the packet buffer address and character count in the I/O Register area. The FE acknowledges the request by first clearing the request in the I/O Register area and then fetching a copy (via DMA access) of the packet into its local data memory. When the fetch is complete the FE informs the SBC, via an interrupt, that it has copied the packet and will transmit it at the next opportunity. When the FE completes the packet transmission, either successfully or unsuccessfully, the SBC is again informed via an interrupt. It should be noted that transmitting a packet is not a single action but consists of multiple actions carried out in their proper sequence. In between each of the individual actions, control is returned to the sampling operation to detect other events, one of which may allow the next individual action in the sequence to begin. To load a received data packet, the SBC places the address of a receive buffer in the I/O Register area and requests a read. The FE copies the buffer address and acknowledges the request by clearing it in the I/O Register area. Some time later when a data packet is received by the FE it is copied into the SBC receive buffer. The FE places a received data packet message along with a character count in the I/O Register area of the SBC's memory and alerts the SBC, via an interrupt, that a message is in the I/O Register area.

In sampling the network, an event that may occur is the receipt of a packet from another node. When a packet has been received by the hardware, both copies (CW and CCW) are available as are the associated error bits. The various error conditions are checked and a good copy of the received packet is selected

for processing. The packet type and address information is checked for validity, and then the appropriate packet handling action is invoked. Finally the packet type is checked against the current state of the node to make sure that this packet type is acceptable at this time. If any conditions occur that cause the packet to become unprocessable, then the appropriate error handling action is invoked.

Each packet handling action does two things. First information is prepared to alert the SBC of the event that just occurred. Second, the appropriate response packet, if any, is formulated and made ready for transmission. The alert information is prepared and copied into the I/O Register area of the SBC's memory and an interrupt is initiated causing the SBC to process this information. The FE then enters its transmit-transfer action. This is the only time-critical action in the FE code, since it must maintain the transmit synchronization timing of the CROSSFIRE loop. In this action the FE is transmitting the response packet onto the CROSSFIRE loop, while simultaneously (interleaved) transfering the received DATA or XID packet to the I/O Register area of the SBC's memory. When the transmit-transfer action completes, information on the results of these operations is copied into the SBC's memory and an interrupt is initiated causing the SBC to process this information. The information on the transmit contains what was transmitted and to whom. The information on the transfer contains the number of characters placed into the current packet receive buffer. Following the transmit-transfer action the sampling operation is again invoked.

In sampling the timers, the events that may occur are errors on packet reception, and no response to a previous transmission. Timers associated with packet reception are common to all nodes and detect different types of receive errors. When these errors are detected they are handled as discussed above. The timers used to indicate no response to a previous transmission is associated with a Primary. When the timeout occurs the Primary consults its internal node database to evaluate the implication of no response. If the database indicates that node has never been up, the assumption is that it does not exist. The current interaction with this node is terminated, and at some later time it is polled again to determine its existence. If the database indicates that node has been up and this is the first time it did not respond then the assumption is that it just failed or is totally out of communications. As a result current interaction is terminated and the SBC is alerted to this new status condition, and at some later time it will be polled again to determine its status. If, on a later poll, that node is discovered to be operational, its new status is reported to the SBC and normal interaction is resumed.

IV. FIBER OPTICS

Design Considerations

Fiber optic components were chosen to meet the objective of maximum distance between nodes, with the restriction that the products chosen must be commercially available. The objective of maximum distance means minimizing losses, not only in the fiber but through other components such as switches, splices, and connectors. The problem of pulse dispersion, which tends to limit bandwidth and/or distance in many systems, was not a serious consideration here because of the low (1 Mbit/second) data rate. The loss considerations, however, are accentuated by the requirement that the fiber optic system must operate over two inter-node lengths since our design rules state that loop integrity must be maintained even when one node has had complete electrical failure. How the above criteria affected the selection of each component will be explained in the following paragraphs.

The fiber selected is graded index, multimode, with core/cladding diameters of 50/125 micrometers respectivly. At a wavelength of 1300 nanometers this fiber has an attenution of 1 dB/km. This type of fiber is readily available in cabled form since it has been used extensively by the telecommunications industry. This brings the added advantage of availablility of connectors and other hardware as well as pigtailed sources and detectors. Single mode fiber, which has even less attenuation, is also readily available. This type of fiber, however, has a very small core size which makes the problems of connectorizing, splicing, and switching very severe, thus it was considered impractical for this application.

For practical use the fiber must be put in cable form by adding strength members and protective materials. Since GRIDNET requires a dual loop, a two-fiber cable was a natural choice. Each fiber is contained in a 1 mm buffer tube with each tube surrounded by Kevlar fibers and then enclosed in a protective sheathing. The two sheathed fibers are then molded together much in the same manner as common lamp cord. The cable was purchased in 2 kilometer lengths which is the maximum standard reel size.

"Loose tube" construction was specified for this cable. This refers to the buffer tube mentioned above which surrounds the fiber. Having the fiber loose in the tube instead of integrally molded to it allows the fiber to move when it expands or contracts due to temperature changes. This helps to preserve the attenuation characteristics over a wide temperature range. It should be noted however that the cable described above was purchased for indoor, laboratory use. Cable for direct burial or even for pulling through building ducts would have to be specified differently.

One of two light sources may be used depending on the

inter-node distance. For long inter-node distances a semiconductor laser is required, while light emitting diodes (LED) may be used for short distances. The laser used is a GaInAsP/InP injection laser diode that is able to launch 2 mw of power into a 50/125 fiber. It comes in a 14-pin dual in-line package with internal thermo-electric cooler, thermistor, and photo-diode monitor detector. Stabilization of the laser is done with external circuitry connected to the cooler, thermistor and detector pins. The laser comes from the factory with a pigtail of buffered fiber attached. The LED that is used for the short inter-node lengths is packaged identically to the laser. Power into the attached pigtail is specified to be 40 microwatts at the rated maximum current of 150 milliamperes.

The detectors are p-i-n type photodiodes with a field effect transistor (FET) preamplifier circuit enclosed in the same package. The combination results in a pinFET receiver with a rated sensitivity of -50 dBm for data rate of 2 Mbaud and an error rate of one in ten to the ninth. Like the source devices, the receiver is housed in a 14-pin dual in-line package and comes with a fiber pigtail attached.

The switches used for optically by-passing a failed node employ optical elements which act as slide-mounted mirrors for coupling light from input fibers to output fibers. Actuation is by a drive coil against a spring. When actuated, two input fibers are aligned with two output fibers to complete optical paths from cable to receiver and transmitter to cable. When power is removed from the device (by-pass mode) the received light is routed back to the outgoing fiber, excluding the node from the optical loop. Two switches are required for each node, one for the clockwise loop and one for counterclockwise. Insertion loss is stated as 3 dB in the active position and 1.8 dB in the by-pass position.

A number of 2 km fiber sections may be needed to span the distance between the source and detector of adjacent nodes, hence it is important to minimize the insertion loss caused by joining each section. Two methods are available: connectorizing and splicing. Connectors exhibit the greater loss so they are used sparingly, namely at the interface between cable and chassis; the convenience of being able to easily disconnect at this point overrides the loss consideration. Interconnection of pigtailed components, such as switch to LED or pinFET, is done with a splice; joining of one reel of cable to another is also done with splices.

A "dry splice" technology was selected for this application, in which the ends of the fibers to be joined are butted together inside an elastic sleeve which contains V-grooved inserts that keep the fibers aligned. This type of splicing requires less expensive equipment than fusion splicing and has the added advantage that the process is reversible, i.e., the components spliced together can be removed. This is an important

consideration for a prototyping situation. Typical insertion loss for the splice is about 0.2 dB. Good performance requires that the ends of the fibers are well cleaved; the cleave should be at a right angle to the axis of the fiber and the end face clean and without lips or hackles. A special hand tool is used which scores and breaks the fiber with controlled force to produce a clean cleave. No other special tools are needed to assemble the splice, but a microscope for inspection of the fiber ends before insertion is a valuable aid. The splice hardware for loose tube fibers is slightly different from that for tight buffered fibers because of the strain relief features of the housing. Hybrid splices are available for joining the two types of fibers.

Expanded beam lens connectors are used for attachment of cables to the fiber optic chassis at each node. This type of connector uses a miniature lens to expand the light beam from the fiber; an identical lens in the mating connector is used to refocus the light on to its fiber. The purpose of beam expansion is to make the alignment tolerances at the mating surface less critical, thus producing a reliable connection with minimum loss. A loss of 0.7 dB is a typical figure for this connector. The mating of fiber to lens is similar to the splice described above in that a well cleaved fiber is inserted into an elastomeric guide mounted on the back side of the lens.

After selecting the major components a loss budget can be calculated which will determine the maximum inter-node distance. Figure 7 summarizes this step in the design process. The worst case configuration occurs when a node fails and communication must take place over twice the normal link distance without regeneration. Under these conditions the optical signal must pass through two lengths of cable, four connectors, one by-pass switch in the fail-safe position, two switches in the actuated position, and a number of splices that connect cable segments and pigtailed components. With a 1.0 milliwatt laser source (0 dBm) and a -50 dBm receiver sensitivity, there is a 50 dB loss budget. Comparing this with the 46 dB loss shown in figure 7, a span of 32 kilometers can be accomodated.

Note that only two links in the phase I loop are the maximum length of 16 kilometers; this is enough to demonstrate the worst case condition described above. The remaining segments of the loop are arbitrarily short, typically a few meters. LED sources can be used for the short lengths for a considerable saving in component costs. Delays can be introduced into these signal paths of the ptototype to simulate the propagation delay that would normally occur in the longer cables.

Integration into GRIDNET

The Fiber Optic Interface for each node is contained in a rack mountable chassis; it consists of a power supply,

electronics board, by-pass switches, and connectors, both electrical and optical. LED sources and pinFET receiver modules mount on the electronics board which also contains drive circuits for the LED's and switches and additional circuitry for the receivers. Nodes that require laser sources are equipped with a separate plug-in type instrumentation chassis for the transmitter and receiver modules.

Standard TTL interface chips are used to drive the LED sources and the by-pass switch coils. The LED is connected in parallel with the driver output transistor in order to provide fast turn on of the LED and to reduce the load transients on the power supply lines. A gate circuit on the front of each by-pass driver provides a means for selected failure conditions in the node to cause by-passing, in addition to the obvious condition of catastrophic power loss.

The output of the pinFET module is a signal whose amplitude is proportional to the light received through the fiber; it varies from a few millivolts to about 4 volts, peak-to-peak, with a dc offset of -1.5 volts. In order to get a TTL compatible signal to send to the FE board the pinFET signal is sent through an operational amplifier, a comparator and a line driver. In order to avoid dc drift problems at the comparator the pinFET signal is ac coupled to the operational amplifier, making it possible for the reference point of the comparator to be ground. The comparator has positive feedback to give it some hysteresis, otherwise it is likely to oscillate or respond to small noise spikes. Since an increase in feedback reduces input sensitivity, the size of the feedback resistor must be chosen carefully for the required dynamic range.

Receiver sensitivity and noise must be considered together because the presence of the latter affects the practical range of the former. Crosstalk may be considered one form of noise. It occurs in the optical signals of the by-pass switches and in the electrical signals on the electronics board. These and other noise sources that are present in the electrical domain must be dealt with by filtering, isolation and care in wiring practice and layout. Putting clockwise and counter-clockwise electronics together on one wire-wrap board, as was done for the short links, is not acceptable for a high sensitivity design. For the long links, receiver circuits were put on separate boards and constructed with special attention to layout and power distribution. Power to these boards is supplied by a high quality linear supply with a separate set of regulators and filters for each board.

V. TIME DISTRIBUTION SYSTEM

The Time Distribution System was designed to be used for collection of performance statistics on the prototype network. It

would not be required in an operational system. It consists of a Master Clock Board installed in the host computer and Time Code Boards in each of the nodes. Distribution of clock pulses and synchronized reset signals is accomplished on a pair of cables which are NOT a part of the GRIDNET communications link. In each node software reads the count (timestamp) from its Time Code Board and stores it in local memory. At some convenient time, the set of timestamps and its associated data is collected by the network control center over GRIDNET.

The design goal for the Time Distribution System was to provide signals for up to 100 nodes. The output line drivers on the Master must be able to drive not only the 100 Time Code inputs but also a substantial length of cable. Furthermore, the lines must be free of noise that would cause false counts and the system must be kept free of ground loop problems. The above considerations led to the decision to use coaxial cable for the signals and to provide isolation transformers for them on each Time Code Board. Further details of driving and receiving techniques will be given below in board descriptions.

The clock rate is 100 kHz which gives the timestamp a resolution of 10 microseconds. The Time Code Boards can accumulate a count of 32 bits for a total time span of nearly 12 hours.

Master Clock Board

The Master Clock Board provides a stable 100 kHz Clock signal for the time code counters, and a master Clear signal which is used to initialize those counters simultaneously. High current line drivers couple each of these signals to coaxial cables for distribution to the GRIDNET nodes. Control of these signals by the host computer is accomplished via an interface to the IEEE-796 bus.

The 100 kHz Clock is derived from a 2 MHz crystal oscillator. The arrival of a clear/start command from the computer causes the oscillator pulses to be gated to the Clock output and a momentary assertion of the Clear output. The Clock runs until a stop command is issued by the computer. Since the Clock and Clear lines are transformer coupled to the Time Code Boards, it is more desirable to transmit bursts of high frequency pulses on these lines rather than the baseband signals. This is accomplished by gating the 2 MHz oscillator output with the 100 kHz Clock signal and the 10 microsecond Clear pulse before sending these signals to their respective line drivers.

Time Code Boards

The Time Code Boards provide a 32-bit timestamp which can be read and stored at each GRIDNET node. Each Time Code Board

accepts the two signals, Clock and Clear, from the Master Clock via coaxial cables. Both of these signals are transformer coupled into a dual line receiver. Since the signals arrive as 2 MHz bursts, they each pass through an "envelope detector" after coming out of the receiver; this converts the Clock signal to a 100 kHz square wave and the Clear signal to a single active-low pulse. These signals are then applied to the appropriate inputs of the counters.

The circuits used for counting Clock pulses are synchronous 8-bit counters; four chips are cascaded to produce a 32-bit counter. When the Clear signal is held low, all counter stages are reset to zero on the positive transition of the Clock; when Clear is high each rise of the Clock increments the count.

The outputs from the counters are strobed into latches on every downward transition of the Clock, except when a data transfer operation is in progress. The reason for this is as follows: The computer can read only 16 bits of the count in one operation, so once the read sequence starts the data must be "frozen" until both halves are read, otherwise the reading may be ambiguous. The latch outputs directly drive the IEEE-796 bus when they are enabled during a read operation.

In addition to being counted, the Clock pulses are monitored by a missing pulse detector circuit. The node processor is notified via an interrupt line if one or more clock pulses are lost, thus alerting the system to the erroneous count.

VI. GRIDNET Graphics

The GRIDNET graphics configuration package is an interactive graphics software program. Its function is to create, modify, and operate on graphical representations of GRIDNET configurations. These configurations can be stored on disk for later modification, display, or processing as well as for use by other programs. The package currently has two major capabilities: (1) to edit a GRIDNET configuration (i.e. create, display, store, and modify) and (2) to simulate and display routings in a GRIDNET configuration. The GRIDNET graphics configuration package is coded in the "C" programming language and it calls routines in the MEGATEK "WHIZZARD 6200" graphics subroutine library. It has been implemented successfully on both a Sun and a VAX running UNIX BSD 4.2 attached to a MEGATEK color graphics workstation.

The GRIDNET Graphics program is organized as a number of modules. The main program is a small control module which serves as a driver for the package. Dependencies on the main program are kept to a minimum, so that other drivers which call routines in the package can be designed without much overhead.

An edit module handles editing of a GRIDNET configuration. A global structure stores the node operational status, the loop breakage status, and graphic ID of the display image for every node and loop. The node operational status indicates the state of each node, either up, down, or nonexistent. The loop breakage status indicates the state of each loop between every node, either complete breakage, clockwise break, counterclockwise break, or no break. The graphic ID is used to reference each display image and perform various graphical manipulations on them.

A routing module simulates routing through the specified GRIDNET configuration and displays the graphic representation of that routing. An option allows the specification of dynamic outages (i.e., outages that occur during the routing) and the simulation and display of the alternative routing that would occur due to these outages.

An interface module handles user interaction, as well as input and output to disk file storage. User interaction occurs through the keyboard and the joy stick (an interactive pointing device). Disk files are used to store various GRIDNET configurations which can be retrieved for display and modification at some later time.

The remaining modules are low level modules concerned with the details of the graphics. A symbol module defines all graphic subroutines, such as loop outlines and Gateway symbols. An instance module handles simple instances of these symbols. A mapping module handles the graphics numerical mappings, such as indexing, generation of screen coordinates, interpreting of screen coordinates, etc. A definition module assigns constants to variables. This is necessary due to the interface to the MEGATEK WHIZZARD software written in FORTRAN (and its required call-by-reference passing of parameters).

VII. Summary

The design of the phase I prototype of the GRIDNET wide area communication network has been presented. The hardware, software, and micro-code of this design were successfully implemented and integrated into an operating CROSSFIRE loop. The Link and Physical layer protocols for inter-node communication were the desired the ADCCP protocols. Separate communication with a development system was established to download code and access CROSSFIRE loop information. This achievement satisfied phase I objectives 1 and 2. Operational performance of this CROSSFIRE prototype demonstrated the ability to both detect and locate outages. This achievement satisfied phase I objectives 3 and 4.

VIII. References

[1] Moore, R. T., Geer, N. F., Graf, H. A., "GRIDNET: An Alternative Large Distributed Network", IEEE Computer, Vol. 17, No. 4, Apr. 1984, pp 57-66.

[2] Moore, R. T., Holt, A. L., Koenig, A. L., Mink, A., and Nacht, G., "Simulation of The Gaurd Control Station in a Computerized Site Security Monitor and Response System", National Bureau of Standards, NBSIR 82-2656, Jan. 1983, NTIS, Springfield, Va 22161.

[3] Moore, R. T., "HYBIRD GRIDNET Packet and Circuit Switching in a Single Network", National Bureau of Standards, NBSIR 82-2588, Oct. 1982 Springfield, Va 22161.

[4] Moore, R. T., "GRIDNET", National Bureau of Standards, NBSIR 80-2149, Oct. 1980, NTIS, Springfield, Va 22161.

[5] Moore, R. T., Carpenter, R. J., Holt, A. L., Koenig, A. L., and Warnar, R. B. J., "Phase II Final Report Computerized Site Security Monitor and Response System", National Bureau of Standards, NBSIR 79-1725, Mar. 1979, NTIS, Springfield, Va 22161.

[6] Epstein, J. A., "A Discussion of GRIDNET Algorithms and Simulation Results", National Bureau of Standards, NBSIR 83-2660, Nov. 1982, NTIS, Springfield, Va 22161.

[7] American National Standards Institute, "American National Standards for Advanced Data Communication Control Procedures (ADCCP)", ANSI X3.66-1979, American National Standards Institute, 1430 Broadway, New York, N.Y. 10018, 1979.

[8] desJardins, R., "Overview and Status of the ISO/ANSI Reference Model of Open System Interconnection", Proc. of Fall CompCon 1980: distributed processing, Wash., D.c., Sept 1980, pp 553-557.

CROSSFIRE DUAL LOOP CONFIGURATION

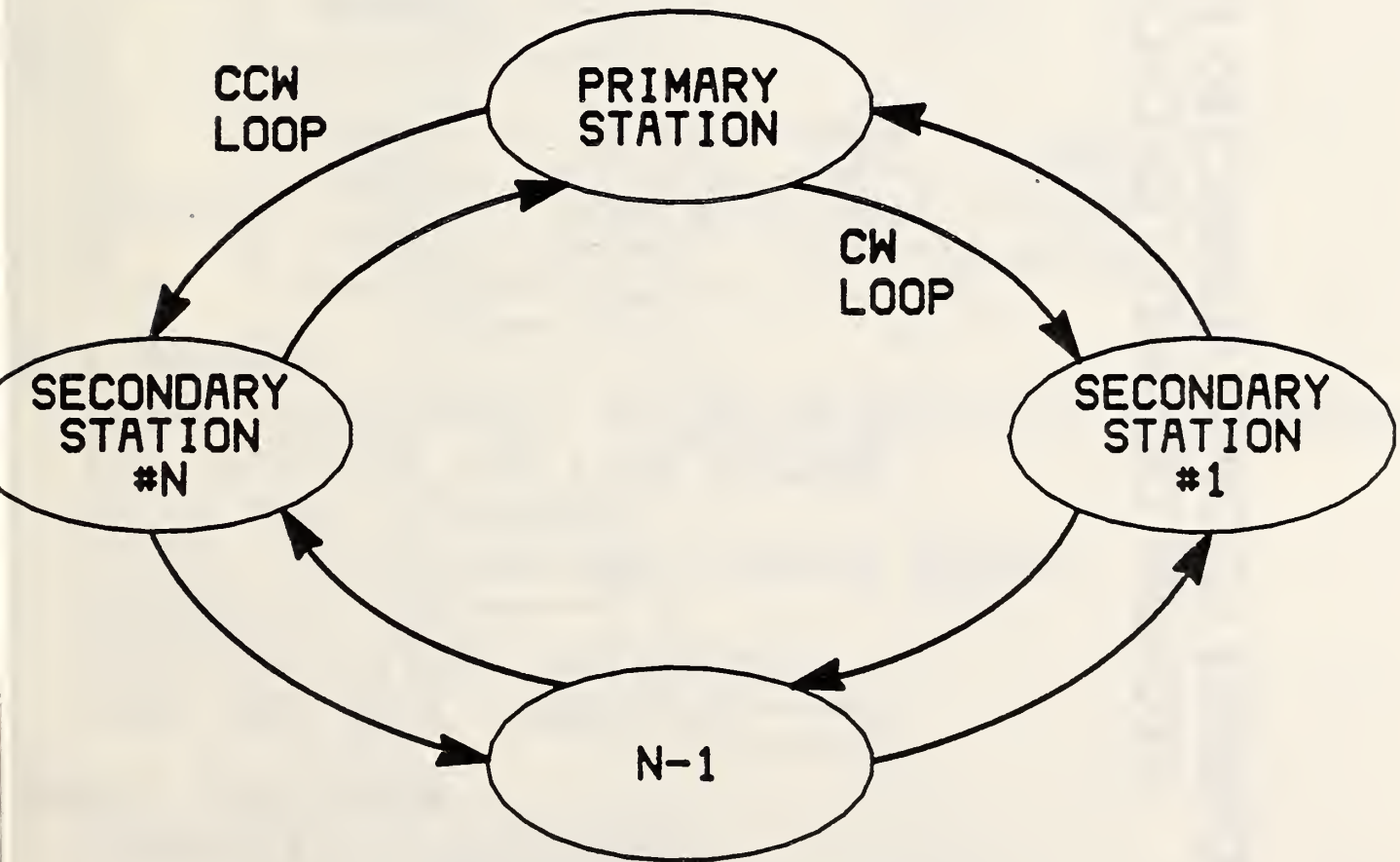


FIGURE 1

CROSSFIRE LOOPS CONNECTED TO FORM REGULAR GRIDNET TOPOLOGY

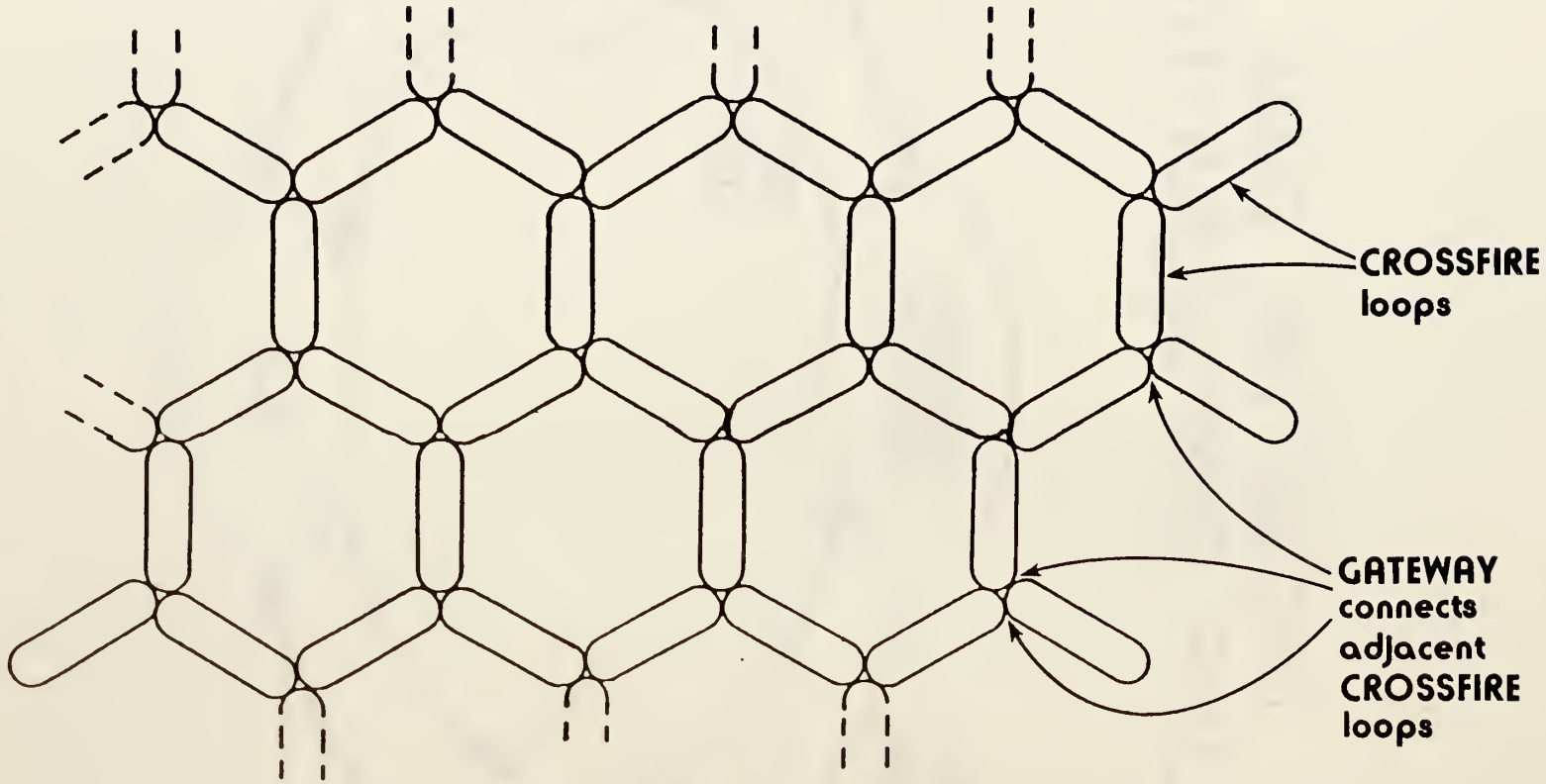
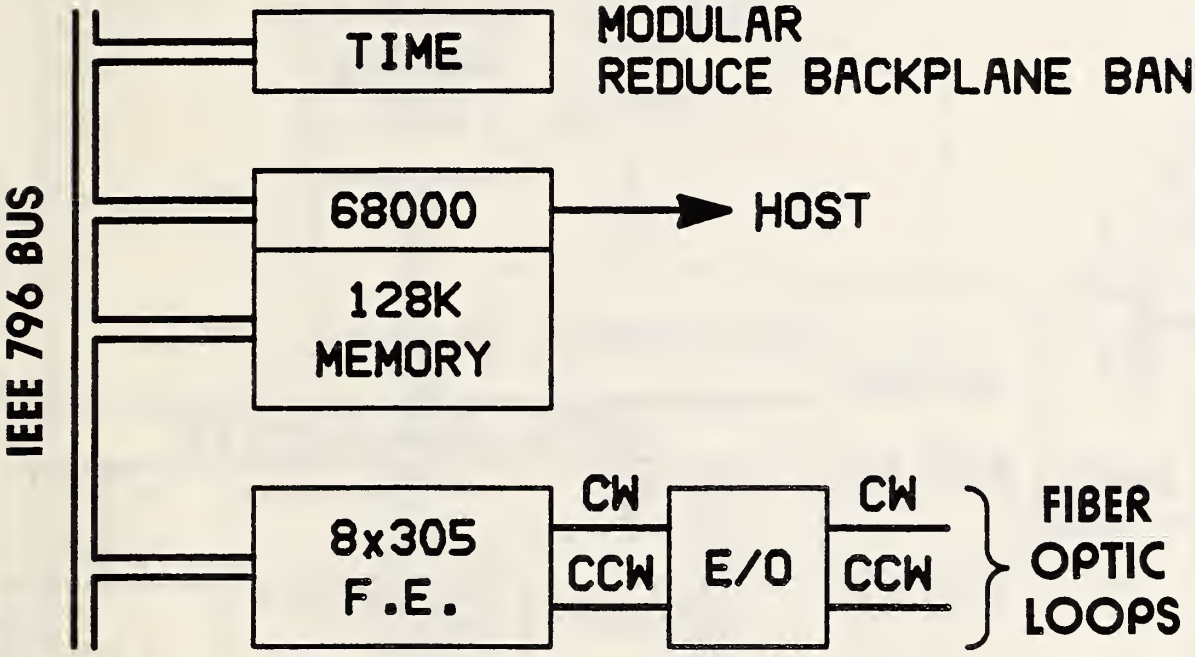


FIGURE 2

NODE

MINIMIZE CUSTOM DESIGN
ISO PROTOCOLS
1 MHz DATA RATE
MODULAR
REDUCE BACKPLANE BANDWIDTH



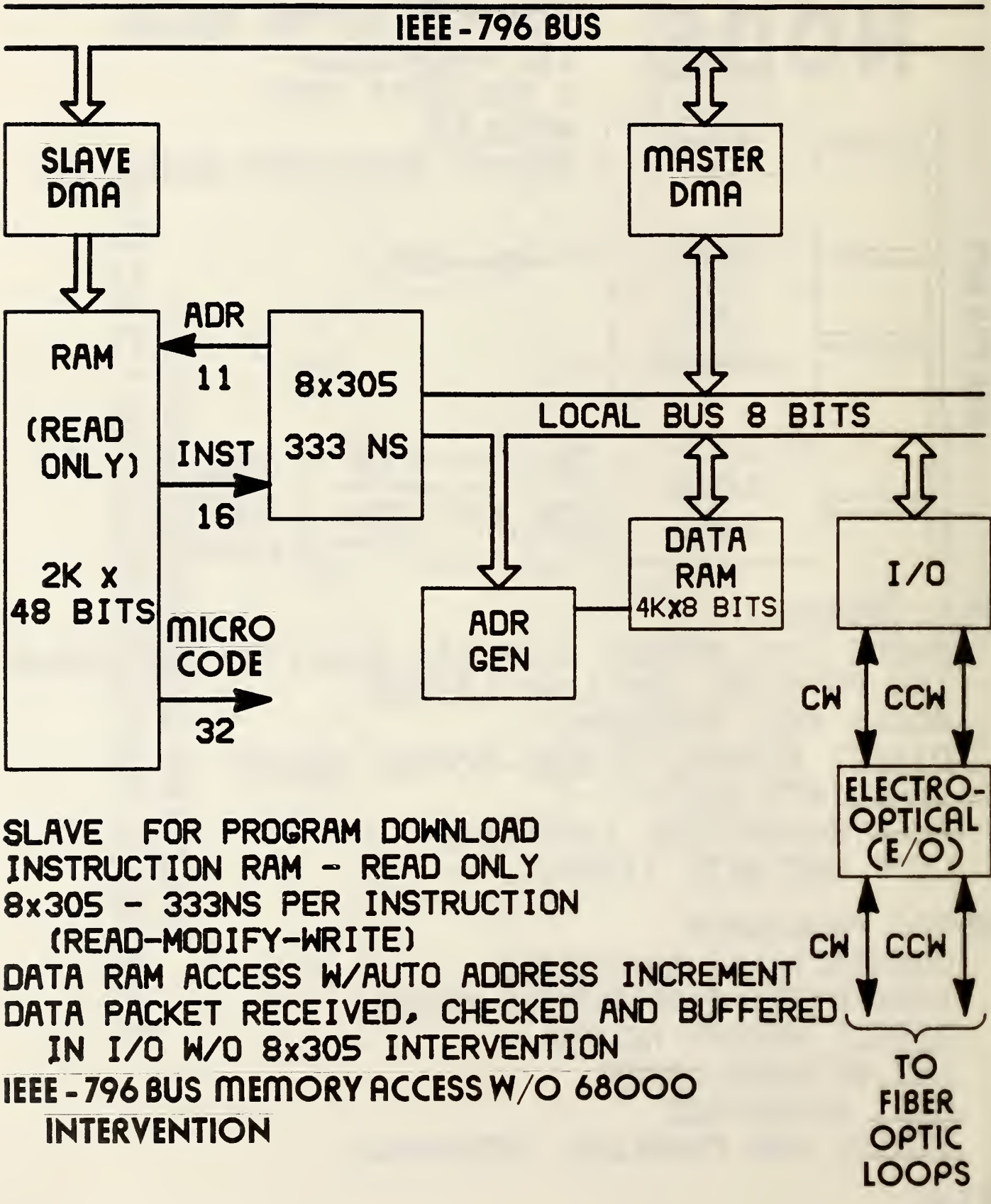
F.E. BOARD
- 8x305 HIGH SPEED, BI-POLAR SCHOTTKY PROCESSOR
- ISO PHYSICAL AND LINK LAYERS
- ADCCP ANSI STANDARD
- DIRECT ACCESS TO DUAL-PORTED MEMORY
- IMMEDIATE ACK'S
- ELECTRO-OPTICAL INTERFACE
- LOOP AND BYTE TIMERS

68000 PROCESSOR
- COMMERCIALLY AVAILABLE
- 128K-BYTES DUAL-PORTED MEMORY
- DIRECT MEMORY ACCESS
- ISO NETWORK LAYER
- HOST INTERFACE
- SERIAL AND PARALLEL INTERFACE

TIME
- STATISTICS FOR PROTOTYPE

FIGURE 2

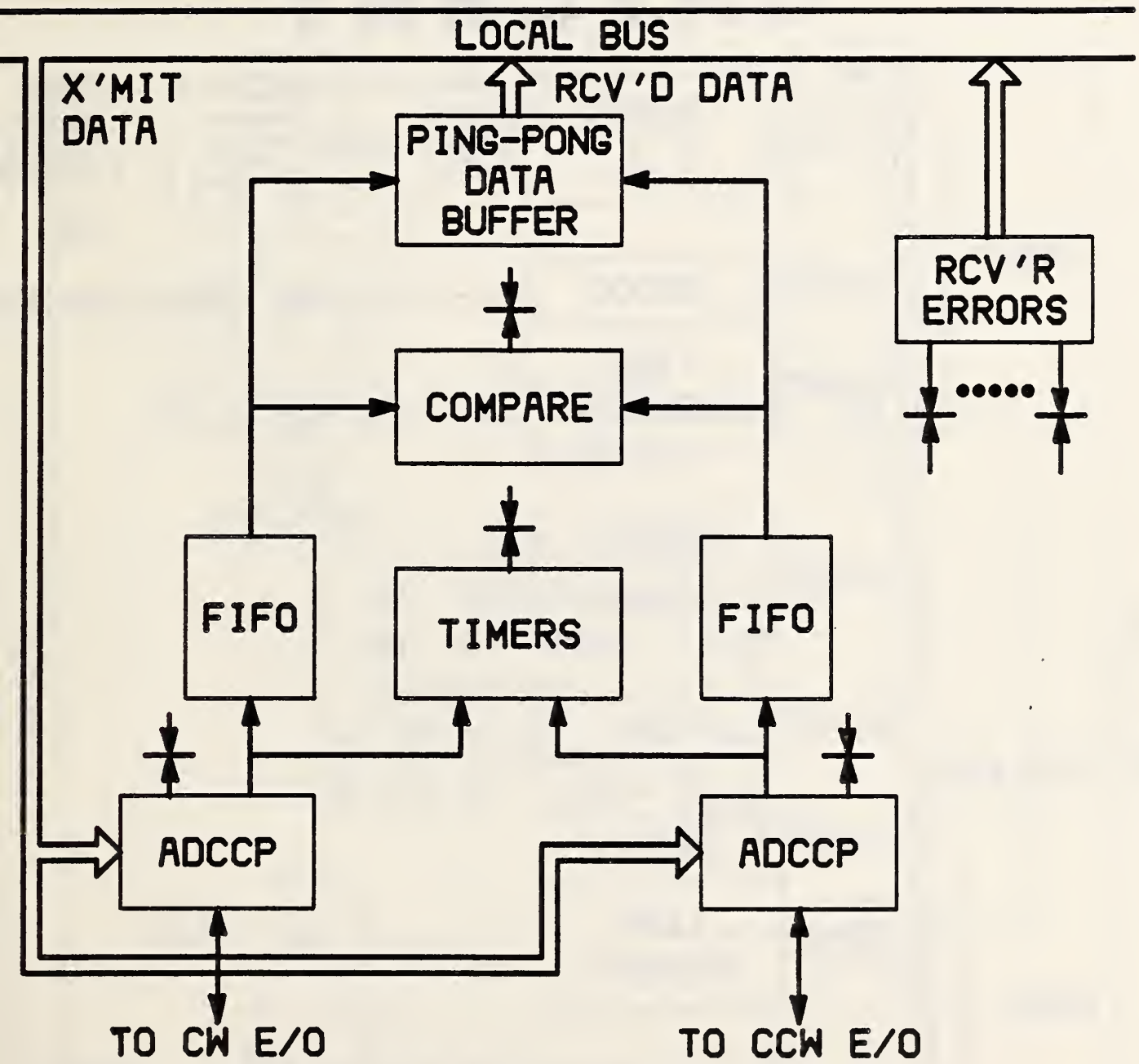
F.E. BOARD



SLAVE FOR PROGRAM DOWNLOAD
INSTRUCTION RAM - READ ONLY
8x305 - 333NS PER INSTRUCTION
(READ-MODIFY-WRITE)
DATA RAM ACCESS W/AUTO ADDRESS INCREMENT
DATA PACKET RECEIVED, CHECKED AND BUFFERED
IN I/O W/O 8x305 INTERVENTION
IEEE-796 BUS MEMORY ACCESS W/O 68000
INTERVENTION

FIGURE 4

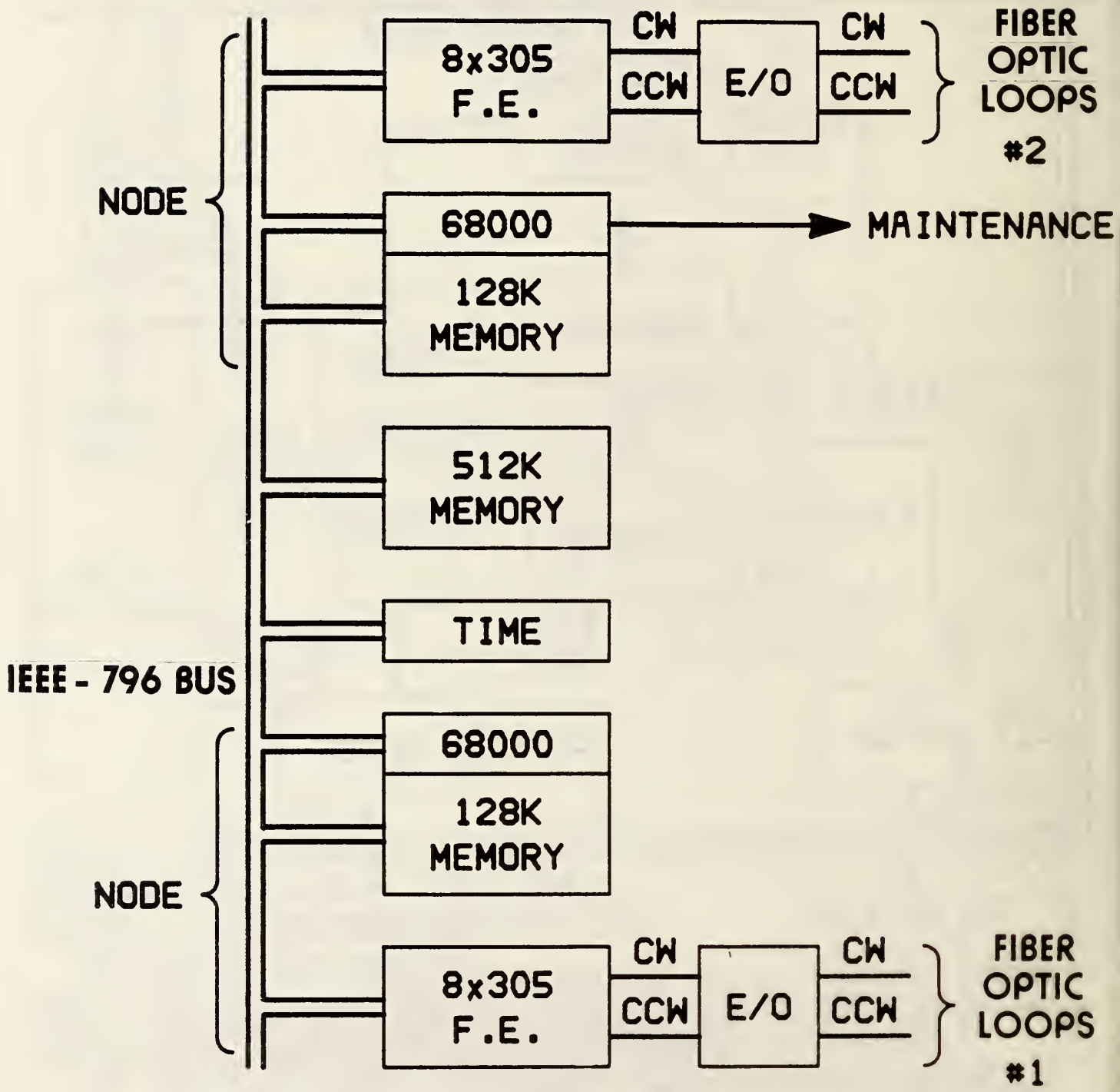
I/O



DATA PACKET = 1K-BYTES MAXIMUM
DATA BUFFER: 2K LOCAL BUS ACCESS WHILE
OTHER 2K RESERVED FOR FIFO'S
BITWISE COMPARISON + CRC
FIFO TO SYNCHRONIZE LOOP DELAY
LOOP DELAY TIMEOUT
BYTE DROPOUT DETECTION
ADCCP PROTOCOLS

FIGURE 5

GATEWAY



2 IDENTICAL NODES
SHARED MEMORY
COMMON BACKPLANE
SINGLE TIME REFERENCE

FIGURE 6

PHASE I LOOP

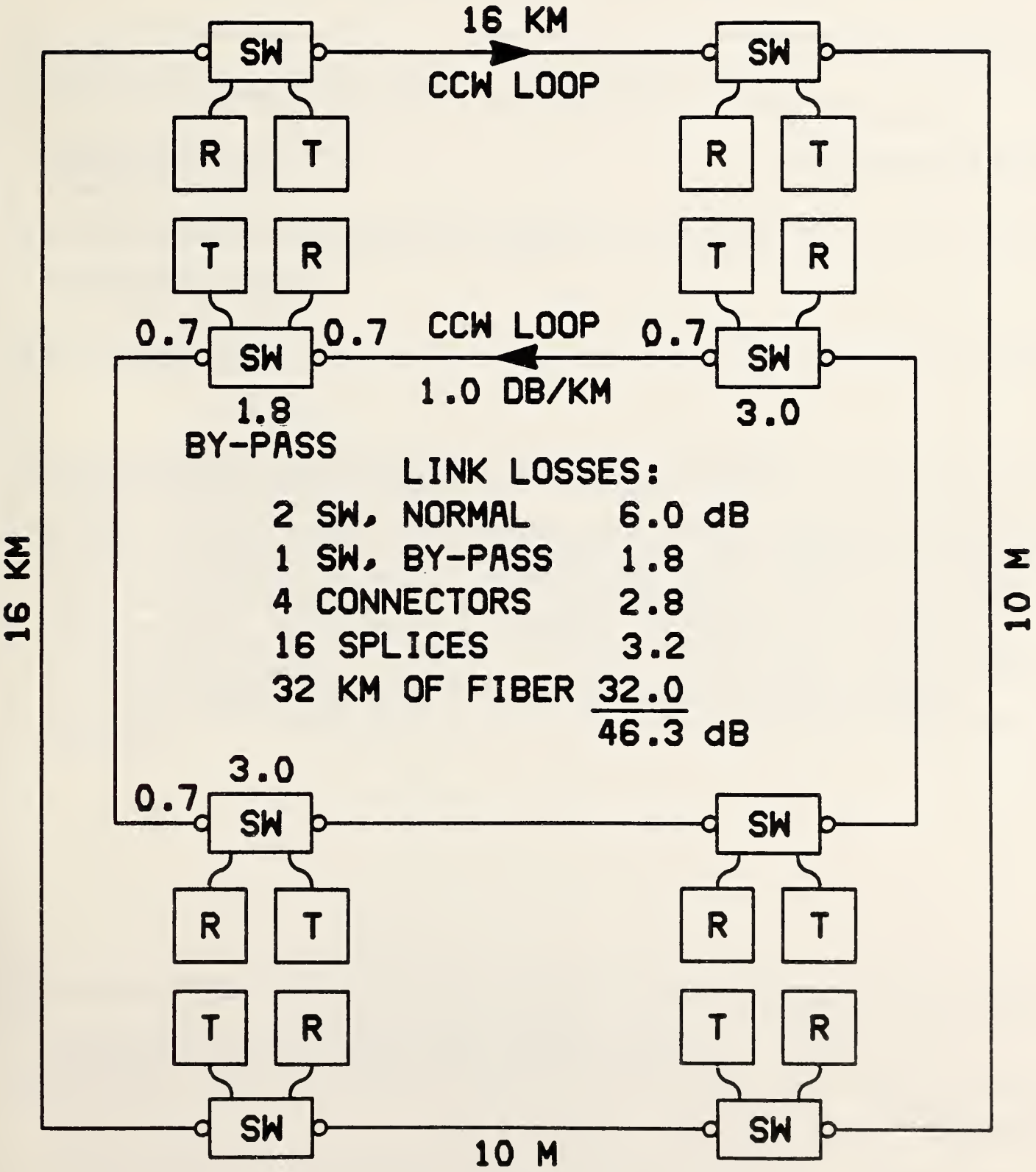


FIGURE 7

NBS-114A (REV. 2-8C)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET (See instructions)	1. PUBLICATION OR REPORT NO. NBSIR 86-3361	2. Performing Organ. Report No.	3. Publication Date APRIL 1986

4. TITLE AND SUBTITLE

GRIDNET: A HIGHLY SURVIVABLE DIGITAL COMMUNICATIONS NETWORK; FINAL REPORT, PHASE I

5. AUTHOR(S)

Alan Mink, George G. Nacht, Alfred L. Koenig, Arthur W. Holt

6. PERFORMING ORGANIZATION (If joint or other than NBS, see instructions)

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, D.C. 20234

7. Contract/Grant No.

8. Type of Report & Period Covered

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (Street, City, State, ZIP)

Defense Nuclear Agency
Washington, DC 20305

10. SUPPLEMENTARY NOTES

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)

GRIDNET is a highly reliable and survivable packet switched, wide area communication network that may consist of thousands of nodes and may span thousands of miles. The reliability of GRIDNET is based on redundant transmission of data via two distinct paths and bitwise comparison of the duplicate received data in addition to error detection codes. The survivability of GRIDNET is attributed to its intrinsic topology, which provides for a number of alternative paths between pairs of nodes. A feasibility prototype of a GRIDNET was proposed as a multi-phase research project. This report describes the design of the phase I GRIDNET prototype which was constructed. This prototype satisfied all of the Phase I operational performance objectives.

12. KEY WORDS (Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)

communications network; fiber optics; hardware design; loop topology; packet switching; protocols; software design; wide area network

13. AVAILABILITY

☒ Unlimited

☐ For Official Distribution. Do Not Release to NTIS

☐ Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.

☒ Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES

33

15. Price

$9.95